Stack cheatsheet for coding interviews

Introduction

A stack is an abstract data type that supports the operations push (insert a new element on the top of the stack) and pop (remove and return the most recently added element, the element at the top of the stack). As an abstract data type, stacks can be implemented using arrays or singly linked lists.
This behavior is commonly called LIFO (last in, first out). The name "stack" for this type of structure comes from the analogy to a set of physical items stacked on top of each other.
Stacks are an important way of supporting nested or recursive function calls and is used to implement depth-first search. Depth-first search can be implemented using recursion or a manual stack.

Learning resources

- Readings
  - Stacks and Overflows, basecs
- Videos
  - Stacks, University of California San Diego

Implementations

Language	API
C++	`std::stack`
Java	`java.util.Stack`
Python	Simulated using List
JavaScript	Simulated using Array

Time complexity

Operation	Big-O
Top/Peek	O(1)
Push	O(1)
Pop	O(1)
isEmpty	O(1)
Search	O(n)

Corner cases

- Empty stack. Popping from an empty stack
- Stack with one item
- Stack with two items

Essential questions

These are essential questions to practice if you're studying for this topic.

- Valid Parentheses
- Implement Queue using Stacks

Recommended practice questions

These are recommended questions to practice after you have studied for the topic and have practiced the essential questions.

- Implement Stack using Queues
- Min Stack
- Asteroid Collision
- Evaluate Reverse Polish Notation
- Basic Calculator
- Basic Calculator II
- Daily Temperatures

- Trapping Rain Water
- Largest Rectangle in Histogram